Online codebook modeling based background subtraction with a moving camera

Liyun Gong
School of Computer Science
University of Lincoln, UK
Email: lgong@lincoln.ac.uk

Miao Yu
School of Computer Science
University of Lincoln, UK
Email: myu@lincoln.ac.uk

Timothy Gordon
School of Engineering
University of Lincoln, UK
Email: tgordon@lincoln.ac.uk

***Abstract*—This paper proposes a new background subtraction method by a moving camera for the object detection. Key points are firstly extracted and tracked. From the tracking results, spatial transformation relationships for the background scenes in consecutive frames are obtained while the current frame is warped to the previous image plane for the camera movement compensation. A codebook background model is constructed and updated in an online way by exploiting the full RGB color information, which is used to distinguish the foreground/background regions. Both qualitative and quantitative experimental results show that the proposed method outperforms its counterparts with a better performance.**

***Index Terms*—moving camera, background subtraction,codebook, image matching**

I. Introduction

Currently there is an increasing demanding of for the monitoring of public and private spaces due to the steady increase for different purposes (e.g., safety, healthcare and crime prevention) [1]. Large numbers of cameras are commonly deployed for monitoring and trained people are asked to watch real-time videos through closed-circuit TV (CCTV) systems to find any abnormal events. However, humans observers are not capable of watching many cameras simultaneously and they will be tired after a long time monitoring. For this reason, automated visual surveillance system enjoys wide researches these days.

For an automated visual surveillance system, one of the most important things is to detect moving objects in the monitored environment. As in [2], [3], the background subtraction technique is one of the most successful approaches for the moving object detection. These methods build statistical background models and extract moving objects by finding regions which do not have similar characteristics to the background model. However, they have limitation that they are only applicable with the stationary cameras.

For detecting moving objects with non-stationary cameras, Cucchiara et al. [4] and Robinault et al. [5] propose panoramic background model based methods. Background models corresponding to panoramic images are constructed through image registration and moving objects are segmented based on the constructed panoramic models. Moreover, the key point matching method is adopted in [6] to solve the possible registration errors for more robust object segmentation. However, the panoramic background model based methods need particularly accurate camera motion model and also suffer from stitching error accumulation. Zhang [7] and Thakoor et al. [8] use a dense optical flow based method. Moving objects are detected by comparing the estimated optical flows with the estimated camera motion. However, dense optical flow requires heavy computation and when the camera motions are large, the computations of optical flow usually fail.

Recently, [9], [10] there has proposed a background subtraction method for non-stationary cameras without constructing a panoramic background model. Camera motions are estimated by the key points tracking while the image warping technique is applied to match the current frame with the background model corresponding to the previous frame, for the moving objets detection. Pixel-wise Gaussian distributions are exploited to model the background considering both spatial and temporal information. Compare with the aforementioned methods [4]–[6], it does not need panoramic background models thus problems caused by the stitching error accumulation can be avoided. Compared with the optical flow based methods, it is both time-efficiency and robust to the large camera motions as mentioned in [7], [8].

In this work, a new method is proposed for the background subtraction detection. Similar to [9], [10], key points are extracted and tracked for consecutive frames. Image warping techniques are then applied to match the consecutive frames for the camera motion compensation. However, different from [9], [10] which only exploit intensity information to build a scalar value based Gaussian distribution for the background modeling, the full RGB color information is exploited and a codebook based method as in [11] is implemented online for the background model construction, model updating and background subtraction. The proposed method keeps the advantages of [9], [10] (e.g., low computational costs, avoiding the stitching error problems, etc.) and can obtain more accurate background subtraction results due to the exploitation of the full RGB color information (instead of only the intensity information as in [9], [10]).

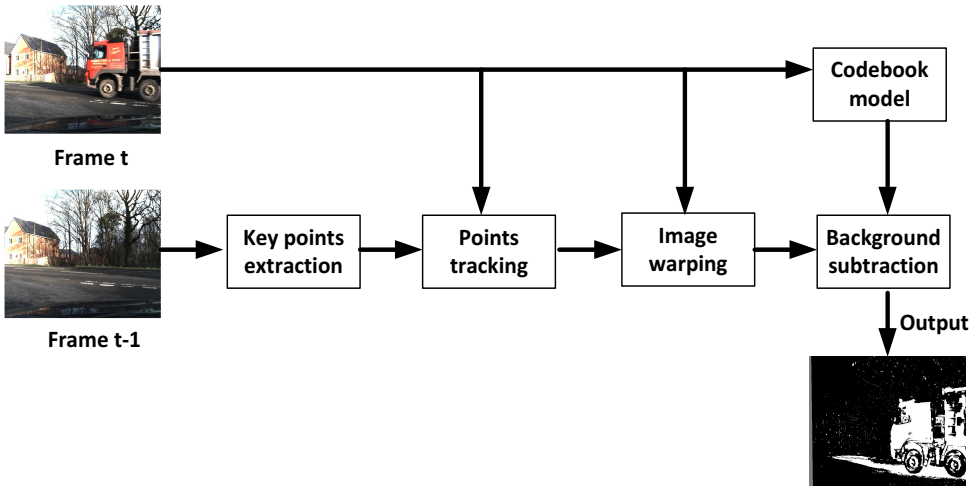

Fig. 1. The flowchart of the proposed algorithm.

The structure of this work is proposed as follows: Section II shows the framework of the proposed approach. Image warping for the consecutive frames matching is proposed in Section III. The methodologies of codebook background model construction and updating, as well as its application for the background subtraction are proposed in Section IV. Experimental results are given in Section V and we give the final conclusions in Section VI.

II. Framework of the Proposed Approach

A flowchart of the proposed algorithm for consecutive frames is presented in Fig. 1. Firstly, key points in a image (e.g., corner points) are extracted and tracked. Certain image warping transformations could then be applied to match the background scenes in consecutive frames, for compensating the background changes due to the camera movement. Based one the warping results, a codebook model is constructed and updated, which is used for the background subtraction to extract moving objects in the current frame. The details of each block are presented in next few sections.

III. Image Warping

Some key points (i.e. corners) are firstly extracted from the image and Lucas Kanade Tracking method (LKT) algorithm [9] is applied to track the key points in consecutive frames, as shown in Fig. 2 (a)–(b). We define $X_{t-1} = [\mathbf{x}_{t-1}^1, ..., \mathbf{x}_{t-1}^N]$ the ensemble of N key points found at time $t-1$ and $X_t = [\mathbf{x}_t^1, ..., \mathbf{x}_t^N]$ is defined as the ensemble of the tracked points at time t. Here $\mathbf{x}_t^i = [u_t^i, v_t^i, 1]$ while u_t^i and v_t^i represent its 2D position in the image. Based on the definitions, we can solve the following equation for obtaining a transformation matrix:

$$X_{t-1} = H \cdot X_t, \tag{1}$$

where the transform matrix H is a 3-by-3 matrix which describes the spatial relationship between two consecutive frames.

As in [9], H can be solved through a least square criteria with:

$$H = X_{t-1} X_t^T (X_t X_t^T)^{-1}, \tag{2}$$

where $(\cdot)^T$ represents the matrix transpose and $(\cdot)^{-1}$ represents the matrix inverse.

By multiplying the estimated transform matrix H on the pixels' positions on the current image, it is warped onto the image plane at the previous time instance and the same background scenes in consecutive frames can approximately overlap with each other, which compensates for the camera movement. One image warping example is shown in Fig. 2 (c)–(d).

IV. Online Codebook Background Subtraction

Certain background model can be constructed for the background subtraction, based on the image warping for aligning the same background scenes in consecutive frames. [9], [10] propose to model the background information using a single Gaussian distribution based on the intensity value; however, it is not realistic to assume a simple Gaussian distribution model for every background pixel (i.e., some background pixels may correspond to more than one color modality [11]); besides, it is not sufficient to only exploit the intensity information instead of the full RGB color information for the background subtraction.

In this work, an online codebook modeling method is developed by exploiting codewords to model the background information in a pixel-wise way. For each pixel, there are two sets of codewords corresponding to it: background codewords set (denoted as $\mathbf{B}_{codewords}$) and cache

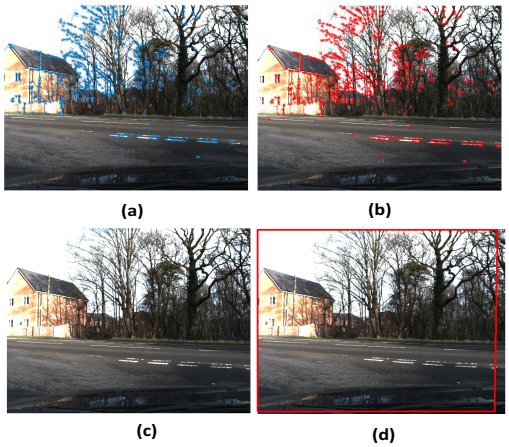

Fig. 2. The illustration of the key points extraction, tracking and image warping. The extracted corner points (blue stars) in a frame and the tracked ones (red stars) are shown in (a) and (b). (c) and (d) show the original frame and its projection on the previous frame plane (enclosed by the red lines) through the image warping.

codewords set (denoted as $\mathbf{C}_{codewords}$). Codewords in $\mathbf{B}_{codewords}$ represent the background color information for that pixel. The ones in $\mathbf{C}_{codewords}$ represent the codewords which are currently inconsistent with the background color, but may potentially be updated to be the background codewords in $\mathbf{B}_{codewords}$ due to the change of the environment.

One codeword denoted as **c** is composed of an RGB vector $\mathbf{v} = (\bar{R}, \bar{G}, \bar{B})$ and a 5-tuple **aux** $= \langle \check{I}, \hat{I}, f, \lambda, p, q \rangle$. The meaning of the elements in the 5-tuple **aux** is shown in Table I:

TABLE I
THE MEANING OF THE ELEMENTS IN THE TUPLE **AUX**.

$\check{I}, \hat{I}$	the min and max brightness of all pixels assigned to this codeword
f	the frequency with which the codeword is matched
λ	the time period that the codeword is not matched
p,q	the first and last matched times of the codeword

A. Codebook model initializing

The codebook model is initialized when the first frame comes. For every pixel in the frame, an associated codeword is constructed with the corresponding **v** being set to be the RGB values for that pixel and **aux** is set to $\{I, I, 1, 0, 1, 1\}$ (where I represents the intensity value), which is put into the background codewords set $\mathbf{B}_{codewords}$. The corresponding cache codewords set $\mathbf{C}_{codewords}$ is set to be empty.

B. Background subtraction

For a newly incoming frame at time instance t, firstly we exploit the matrix H as (2) to project every pixel position to the image plane at $t-1$ according to (1). For a pixel denoted as p_t^i, if its projected position (denoted as $\mathbf{x}_{t-1}^i$) is beyond the image dimensionality (assuming every image has the same dimensionality of $H \times W$), then the pixel belongs to the newly emerging region due to the camera movement and a background codeword is initialized for that pixel.

If $\mathbf{x}_{t-1}^i$ is within the image dimensionality, p_t^i will be compared with codewords in both $\mathbf{B}_{codewords}$ and $\mathbf{C}_{codewords}$ associated with pixels within a small region (denoted as R_{t-1}^i) around $\mathbf{x}_{t-1}^i$ considering possible warping errors, to determine whether p_t^i is a background pixel or not. In this work, R_{t-1}^i is set to be square centering at $\mathbf{x}_{t-1}^i$ with the width being l. Comparisons will be made by both the color distortion measurement and brightness evaluation [11], as illustrated in (3):

$$\begin{aligned} colordist(\mathbf{p}, \mathbf{c}) &= \sqrt{\| \mathbf{p} \|^2 - \frac{\mathbf{p} \cdot \mathbf{v}}{\| \mathbf{v}}\|^2} \\ brightness(\mathbf{p}, \mathbf{c}) &= \begin{cases} true & if \quad I_{low} \leq I \leq I_{hi} \\ false & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

where $\mathbf{p} = [r, g, b]$ representing the RGB vector for a pixel p in a three channels color image and I represents the corresponding intensity value. I_{low} and I_{hi} are calculated from the first two components $\check{I}$ and $\hat{I}$ of the **aux** vector in the codeword **c** as:

$$I_{low} = \alpha \hat{I}, I_{hi} = min\{\beta \hat{I}, \frac{\check{I}}{\alpha}\} \tag{4}$$

where α and β are manually set parameters.

As in [11], it is mentioned that a pixel p is matched with a codeword **c**, if the compared $colordist$ value is smaller than a threshold and $brightness$ is true. For the pixel p_t^i, if there exists a matched background codeword associated with any pixel within R_{t-1}^i, then the pixel is regarded as a background pixel; otherwise, if no such codewords exist, p_t^i is regarded as a foreground pixel representing a moving object.

C. Online background model updating

Codewords in the background model are designed to update in an online way to account for the environmental changes in video streams. For any matched codeword, its corresponding RGB vector **v** and components in the **aux** will be updated by the $[r, g, b]$ vector and intensity value I of the pixel p_t^i as in (5):

$$\begin{aligned} \mathbf{v} &\leftarrow (\frac{f_m R + r}{f_m + 1}, \frac{f_m G + g}{f_m + 1}, \frac{f_m B + b}{f_m + 1}), \\ \check{I} &\leftarrow min\{I, \check{I}_m\}, \\ \hat{I} &\leftarrow max\{I, \hat{I}_m\}, \\ f_m &\leftarrow f_m + 1, \\ \lambda &= 0 \\ q &\leftarrow t. \end{aligned} \tag{5}$$

Fig. 3. Illustrations of the background subtraction results for different scenes: (a). Original frames (b). Ground truth moving objects regions (c). Background subtraction results based on the method in [9], [10] (d). Background subtraction results by the proposed method

For the codewords are not matched, the λ component will be updated as $\lambda \leftarrow \lambda + 1$ and all other components will kept the same. Codewords in the set $\mathbf{C}_{codewords}$ will be added into $\mathbf{B}_{codewords}$ when matched frequently ($f_m > th_1$) indicating the background changes. Besides, codewords with $\lambda > th_2$ will be removed from the codewords set due to not being matched for a certain time period. th_1 and th_2 are preset threshold values.

V. Experimental results

The proposed method is tested on a video sequence for detecting moving objects (people, cyclists, cars and vans) on the road, which is recorded by a camera mounted on a vehicle. Recorded frames have a dimensionality of 512×640. Representative frames are shown as in Fig. 4.

Fig. 4. Selective images in the recorded video sequence.

Fig. 3 shows the qualitative moving object detection results of the proposed background subtraction method, as well as its counterpart in different traffic scenarios. Intuitively, we can observe the background subtraction results by the proposed method could better match the ground truth results with a comparatively smaller number of background pixels being mistaken as foreground ones.

The recall and precision values as in (6) are used for a quantitative comparison. The higher recall and precision values are, the better the background subtraction algorithm is as indicated from this equation. A piece of video sequence is chosen and the moving objects (as illustrated in Fig. 5) are extracted by different background subtraction methods. The related recall and precision values for every frame are calculated and shown in Fig. 6, from which we can see that for most of the time instances the proposed method could obtain both higher recall and precision values, which indicate the better performance of the proposed algorithm. The reason for the better performance of the

Fig. 5. Moving objects (cyclist and vehicle) and the corresponding ground truth regions.

proposed algorithm is that: compared with the state-of-the-art methods [9], [10] which only exploit the intensity information, the proposed methods exploits the full RGB color information for the background modeling; besides, instead of assuming a simple Gaussian distribution for each pixel, the codebook based method can model the background information in a more representative way.

$$precison = \frac{tp}{tp + fp}, recall = \frac{tp}{tp + tn} \quad (6)$$

tp: foreground pixels which are correctly detected

fp: background pixels which are mistaken as foreground ones

tn: correctly detected backgrround pixels

VI. Conclusion

In this work, we propose a novel background subtraction method based on the moving camera. Key points are extracted and tracked while image warping is applied to match the same background scenes for consecutive frames for compensating the camera movement. A codebook model method is constructed and online updated, for extracting the moving foreground objects by exploiting the full RGB color information and codewords based representation. Experimental results on a real video sequence show the advantages of the proposed method over its state-of-the-art counterparts. For future works, we will work on the improvement of the current image warping method. Same background regions in consecutive frames can thus be matched in a better way, which leads to better background subtraction results.

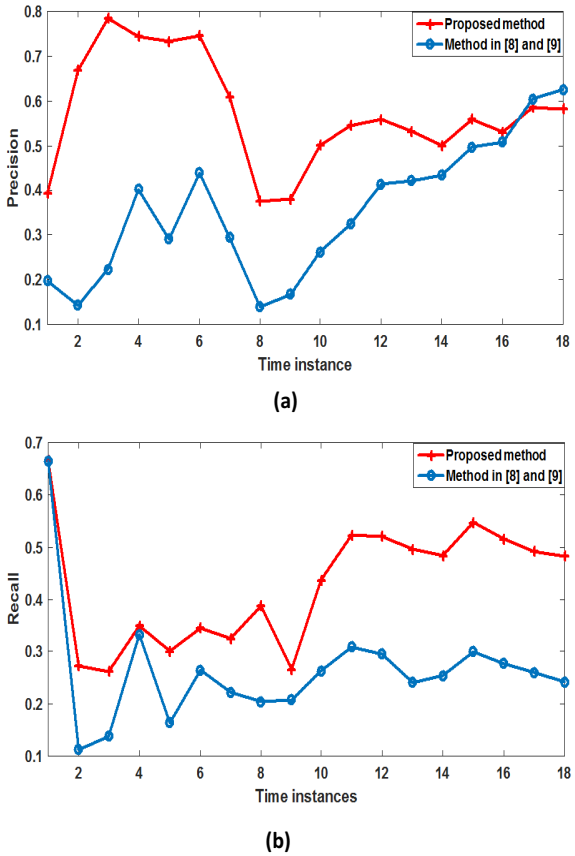

Fig. 6. The precision (a) and recall (b) curves for the background subtraction results on a piece of video sequence.

References

[1] P. Remagnino, S. Velastin, G. Foresti, and M. Trivedi, "Novel concepts and challenges for the next generation of video surveillance systems," *Machine Vision Application*, vol. 18, no. 3, pp. 135–137, 2007.

[2] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceeding of IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[3] T. Ko, S. Soatto, and D. Estrin, "Warping background subtraction," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1331–1338.

[4] R. Cucchiara, A. Prati, and R. Vezzani, "Advanced video surveillance with pan tilt zoom cameras," in *In: Proceedings ofWorkshop onVisual Surveillance (VS) at ECCV*, 2006.

[5] L. Robinault, S. Bres, and S. Miguet, "Real time foreground object detection using ptz camera," in *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, 2009, vol. 1.

[6] C. Guillot, M. Taron, , P. Sayd, Q. Pham, C. Tilmant, and J. Lavest, "Background subtraction adapted to ptz cameras by keypoint density estimation," in *Proceedings of the British Machine Vision Conference*, vol. 34, pp. 1–10.

[7] Y. Zhang, S. Kiselewich, , W. Bauson, and R. Hammoud, "Robustmoving object detection at distance in the visible spectrum and beyond using a moving camera," in *In: Conference on Computer Vision and Pattern Recognition Workshop, New York, USA*, 2006.

[8] N. Thakoor, J. Gao, , and H. Chen, "Automatic object detection in video sequences with camera in motion," in *Advanced Concepts for Intelligent Vision Systems, Brussels, Belgium*, 2004.

[9] S. Kim, K. Yun, K. Yi, S. Kim, and J. Choi, "Detection of moving objects with a moving camera using non-panoramic background model," *Machine Vision and Applications*, vol. 24, no. 5, pp. 1015–1028, 2013.

[10] A. Viswanath, R. Behera, V. Senthamilarasu, and K. Kutty, "Background modelling from a moving camera," *Procedia Computer Science*, vol. 58, pp. 289–296, 2015.

[11] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foregroundbackground segmentation using codebook model," *Real Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.